FRACTIONAL CLOCK DIVIDER USING DIGITAL TECHNIQUES

Field of the Invention

The present invention relates to clock division, and, in particular, to a method of clock division that provides an output clock signal having a clock period that is an integral or fractional multiple of the reference clock period.

Background of the Invention

A clock divider is arranged to provide an output clock signal in response to a reference clock signal. The clock divider is configured to use digital techniques to divide the reference clock frequency by an integer value. For example, a typical clock divider may be configured to generate a 7.5 MHz clock signal by dividing a 60 MHz reference clock signal by 8.

Brief Description of the Drawings

Non-limiting and non-exhaustive embodiments of the present invention are described with reference to the following drawings.

FIG. 1 is an illustration of an example embodiment of a fractional clock divider system;

FIG. 2A is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=5;

FIG. 2B is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=3;

FIG. 2C is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=8;

FIG. 3 is a tabular illustration of example reference values for an example ROM table; and

FIG. 4A and FIG. 4B illustrate an example embodiment of a process of fractional clock division, in accordance with aspects of the present invention.

Detailed Description of the Preferred Embodiment

Various embodiments of the present invention will be described in detail with reference to the drawings, where like reference numerals represent like parts and assemblies throughout the several views. Reference to various embodiments does not limit the scope of the invention, which is limited only by the scope of the claims attached hereto. Additionally, any examples set forth in this specification are not intended to be limiting and merely set forth some of the many possible embodiments for the claimed invention.

Throughout the specification and claims, the following terms take at least the meanings explicitly associated herein, unless the context clearly dictates otherwise. The meanings identified below are not intended to limit the terms, but merely provide illustrative examples for the terms. The meaning of "a," "an," and "the" includes plural reference, the meaning of "in" includes "in" and "on." The term "connected" means a direct electrical connection between the items connected, without any intermediate devices. The term "coupled" means either a direct electrical connection between the items connected, or an indirect connection through one or more passive or active intermediary devices. The term "circuit" means either a single component or a multiplicity of components, either active and/or passive, that are coupled together to provide a desired function. The term "signal" means at least one current, voltage, charge, temperature, data, or other signal.

Briefly stated, the invention is related to a fractional clock divider system and method. The clock divider is configured to provide an output clock signal in response to an input clock signal. The frequency of the output clock signal may be an integral or fractional division of the input clock signal. The output frequency is equal to: (ref_freq * 2) / mul, where ref_freq is the frequency of the input clock signal, and mul is a selected integer that is greater than one. The positive and negative edges of the input

clock are counted to provide a positive count and a negative count respectively. A table is configured to store preselected reference values. A logic circuit is configured to control the output clock signal such that an appropriate clock transition occurs in the output clock signal when the positive and negative count reach the corresponding preselected reference values.

FIG. 1 is an illustration of an example embodiment of a fractional clock divider system (100) that is arranged in accordance with aspects of the present invention. System 100 includes a first counter block (110), a second counter block (112), a ROM block (120), an inverter circuit (130), a decode logic block (140), a reset logic block (150), and a jitter minimizing block (160). An example jitter minimizing block (160) comprises flip-flops (170, 171), a multiplexer (180), and a buffer (190).

System 100 is configured to operate as follows below. Counter block 110 is configured to count positive edges of an input clock signal (CLK), and to provide a positive edge count signal (pcount_d). Counter block 110 is also responsive to a reset signal (reset) and a positive edge counter enable signal (p_en). Inverter circuit 130 is configured to invert signal CLK to provide an inverted clock signal (CLK_N). Counter block 112 is configured to count negative edges of signal CLK, and to provide a negative edge count signal (ncount_d). Counter block 112 is also responsive to the reset signal (rst) and a negative edge counter enable signal (n_en).

Reset logic block 150 is configured to provide signals rst, p_en, and n_en in response to signal CLK, signal pcount_d, signal ncount_d, and a mul signal (smul). Signal smul is a digital signal having a value that corresponds to a selected integer (mul). Reset logic block 150 is configured to control counter block 110 such that pcount_d is reset (e.g. to 1) when the next positive edge of CLK occurs after pcount_d reaches pmax (e.g. mul). Reset logic block 150 is further configured to control counter block 112 such that ncount_d is reset (e.g. to 1) when the next negative edge of CLK occurs after ncount_d reaches nmax (e.g. mul). Reset logic block 150 is further configured to assert signal rst for several cycles of CLK to reset counter blocks 110 and 112 when mul is changed. Reset logic block 150 is further configured to assert signal

p_en to enable counter block 110 to resume counting when signal rst is deasserted. Reset logic block 150 is further configured to assert signal n_en to enable counter block 112 to resume counting when the next negative edge of CLK occurs after counter block 110 is enabled. Reset logic block 150 is configured to enable counter block 110 before counter block 112 such that pcount_d is never less than ncount_d before pcount_d and ncount_d reach mul.

ROM block 120 is configured to store a set of preselected reference values (ref) for each value of mul. ROM block 120 is configured to provide the set of reference values (ref) that are associated with mul. Optionally, ROM block 120 may be responsive to signal CLK and a read enable signal (Read). Although FIG. 1 shows that the set of reference values (ref) are stored and provided by ROM block 120, an alternative block may be used to store and provide the set of reference values (ref). For example, the set of reference values (ref) may be stored and provided by a lookup table or a wired logic circuit.

Decode logic block 140 is configured to provide a divided clock signal (CLK_D) in response to signals pcount_d, ncount_d, and the set of preselected reference values (ref). The frequency of signal CLK_D is approximately equal to ref_freq*2/mul, where ref_freq is the frequency of signal CLK.

In one example of a fractional clock divider system (100), signal CLK_D is the output clock signal, and jitter reduction block 160 is omitted from the fractional clock divider system (100). Another example of a fractional clock divider system (100) comprises jitter reduction block 160. Jitter reduction block 160 is configured to reduce the jitter associated with signal CLK_D to provide an output clock signal (CLK_OUT). Jitter reduction block 160 is further configured to filter out the effects of toggling inputs on the output clock signal (CLK_OUT).

Flip-flop 170 is configured to provide signal CLK_H in response to signal CLK_D. Flip-flop 170 is triggered on the rising edge of signal CLK. Flip-flop 171 is configured to provide signal CLK_L in response to signal CLK_D. Flip-flop 171 is triggered on the falling edge of signal CLK. Multiplexer 180 is configured to provide

signal CLKO in response to signals CLK_H, CLK_L, and CLK. Multiplexer 180 is configured such that signal CLKO corresponds to signal CLK_H when the input clock (CLK) corresponds to a first logic level. Multiplexer 180 is further configured such that signal CLKO corresponds to signal CLK_L when the input clock (CLK) corresponds to a second logic level. According to one example, the first logic level corresponds to logic 0, and the second logic level corresponds to logic 1. Buffer 190 is configured to buffer signal CLKO to provide the output clock signal (CLK_OUT).

FIG. 2A is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=5, according to aspects of the present invention. FIG. 2B is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=3, according to aspects of the present invention. FIG. 2C is graphical illustration of a timing diagram for an example embodiment of a fractional clock divider system for the case when mul=8, according to aspects of the present invention.

FIG. 2A-2C illustrate signal CLK and signal CLK_D for different values of mul. Signal CLK_D is divided into phases, where each phase corresponds to two clock cycles of signal CLK_D. Decode logic block 140 is configured to control when each clock edge of signal CLK_D occurs. Phase 1 is the first two clock cycles of signal CLK_D that occur after mul has changed. Point A is the first positive edge of signal CLK_D that occurs in Phase 1. Point B is the first negative edge of signal CLK_D that occurs in each phase. Point C is the first positive edge of signal CLK_D that occurs after Point B in each phase. Point D is the first negative edge of signal CLK_D that occurs after Point C in each phase. Point E is the first positive edge of signal CLK_D that occurs after point D in each phase.

Decode logic block 140 is further configured to cause Point A to occur when pcount_d=1 and ncount_d=0. Decode logic block 140 is further configured to cause Points B, C, and D to occur according to the preselected reference values (ref). Decode logic block 140 is further configured to cause point E to occur when the next rising edge of signal CLK occurs after pcount_d=mul.

FIG. 3 is a tabular illustration of example reference values for an example table, in accordance with aspects of the present invention. Example ROM block 120 is configured to provide preselected reference values (ref) that are associated with mul. An example of ROM block 120 is configured to provide 3 reference pairs: rphl1, rplh, and rphl2. Each reference pair comprises two reference values. Reference pair rphl1 comprises reference high-to-low positive 1 (refhlp1) and reference high-to-low negative 1 (refhln1). Reference pair rphl comprises reference low-to-high positive (reflhp) and reference low-to-high negative (reflhn). Reference pair rphl2 comprises reference high-to-low positive 2 (refhlp2) and reference high-to-low negative 2 (refhln2). Refhlp1 and refhln1 correspond to the values for pcount_d and ncount_d respectively at which point B has been preselected to occur for the current value of mul. Reflhp and reflhn corresponds to the values for pcount_d and ncount_d respectively at which point C has been preselected to occur for the current value of mul. Refhlp2 and refhln2 corresponds to the values for pcount_d and ncount_d respectively at which point D has been preselected to occur for the current value of mul.

The reference values (ref) for each value of mul may be preselected according to the duty cycle desired for signal CLK_D. Generally, it is desirable for the duty cycle to be as close to 50% as possible. The reference values (ref) shown in FIG. 3 are preselected such that the duty cycle of signal CLK_D is as close to 50% as possible and such that the high pulse is longer in duration than the low pulse when mul is odd. Alternatively, the reference values (ref) may also be preselected such that the duty cycle of signal CLK is as close to 50% as possible and such that the low pulse is longer in duration than the high pulse when mul is odd. As another alternative, other reference values (ref) may be preselected such that other duty cycles are attained for signal CLK_D.

FIG 4A and FIG. 4B illustrate an example embodiment of a process (400) of fractional clock division, in accordance with aspects of the present invention. After a start block (402), process 400 proceeds to block 403. At block 403, signal CLK_D is adjusted such that signal CLK_D corresponds to a second logical level (e.g.

low). Processing then proceeds from block 403 to block 404. At block 404, reference values associated with the current value of mul are retrieved from a table. Alternatively, the reference value may be retrieved by another form of memory than a table (e.g. ROM, RAM, PLD, etc.) Processing then proceeds from block 404 to block 406. At block 406, a reset signal is asserted. Pcount_d and ncount_d are reset to an initial value (e.g. zero) while the reset signal is asserted. Processing then proceeds from block 406 to block 408. At block 408, the reset signal is deasserted after a few cycles of signal CLK. Processing then proceeds from block 408 to block 410.

At block 410, signal p_en is asserted when the next rising edge of signal CLK occurs after the reset signal is deasserted. Processing then proceeds from block 410 to block 412. At block 412, signal n_en is asserted on the next falling edge of signal CLK that occurs after signal p_en is asserted. Processing then proceeds from block 412 to block 413. At block 413, process 400 waits for the next positive edge of signal CLK. Processing then proceeds from block 413 to block 414. At block 414, positive edges of signal CLK are counted. Additionally, signal CLK_D is adjusted such that signal CLK_D corresponds to a first logical level (e.g. logic "1") when the first positive edge of signal CLK is counted. Processing then proceeds from block 414 to block 416. At block 416, negative edges of signal CLK are counted when the first negative edge of signal CLK occurs after counting of the positive edges of signal CLK begins. Processing then proceeds from block 416 to block 417.

At block 417, mul is evaluated. Processing then proceeds from block 417 to decision block 418. At decision block 418, a determination is made whether mul has changed. Processing proceeds from decision block 418 to block 403 when mul has changed. Alternatively, processing proceeds from decision block 418 to block 419 when mul is unchanged. At block 419, pcount_d and ncount_d are evaluated. At block 419, pcount_d and ncount_d are compared with the reference values (ref). Processing then proceeds from block 419 to decision block 420. At decision block 420, a determination is made whether pcount_d=refhlp1 and ncount_d=refhln1. Processing proceeds from decision block 420 to block 430 when pcount_d=refhlp1 and

ncount_d=refhln1. Alternatively, processing proceeds from decision block 420 to decision block 421 when either pcount_d≠refhlp1 or ncount_d≠refhln1. At block 430, signal CLK_D is adjusted such that signal CLK_D corresponds to a second logic level (e.g. low). Processing then proceeds from block 430 to block decision block 422. According to one example, decision blocks 420-422 are be performed simultaneously with combinational logic.

At decision block 421, a determination is made whether pcount_d=refhlp2 and ncount_d=refhlpn2. Processing proceeds from decision block 421 to block 430 when pcount_d=refhlp2 and ncount_d=refhln2. Alternatively, processing proceeds from decision block 421 to decision block 422 when either pcount_d≠refhlp2 or ncount_d≠refhln2. At decision block 422, a determination is made whether pcount_d=reflhp and ncount_d=reflhn. Processing proceeds from decision block 422 to decision block 424 when either pcount_d≠reflhp or ncount_d≠reflhn. Alternatively, processing proceeds from decision block 422 to block 432 when pcount_d=reflhp and ncount_d=reflhn. Although decision blocks 420-422 are illustrated as separate blocks, each of the comparisons may be performed at the same

At block 432, signal CLK_D is adjusted such that signal CLK_D corresponds to a first logic level (e.g. logic 1). Processing then proceeds from block 432 to block decision block 424. At decision block 424, a determination is made whether pcount_d has reached pmax (e.g. mul). Processing proceeds from decision block 424 to decision block 426 when pcount_d has not reached pmax. Alternatively, processing proceeds from decision block 424 to block 434 when pcount_d has reached pmax. At block 434, signal CLK_D is adjusted such that signal CLK_D corresponds to a first logic level (e.g. logic 1). Processing then proceeds from block 434 to block 436. At block 436, pcount_d is reset (e.g. to 1) at the next rising edge of signal CLK. Alternatively, pcount_d may be reset to 0 or some other value. Processing then proceeds from block 436 to decision block 426.

At decision block 426, a determination is made whether ncount_d has reached nmax (e.g. mul). Processing proceeds from decision block 426 to decision

block 418 when ncount_d has not reached nmax. Alternatively, processing proceeds from decision block 426 to block 438 when ncount_d has reached nmax. At block 438, ncount_d is reset (e.g. to 1) at the next falling edge of signal CLK. Processing then proceeds from block 438 to decision block 418.

An example of process 400 was illustrated in which pcount_d is always greater than or equal to ncount_d. According to another example of process 400, ncount_d is always greater than or equal to pcount_d. According to this example, the order of signals n_en and p_en are reversed.

The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.